jotto - a word guessing game

Srini, rs@toprllc.com

August 4, 2021

1 Introduction

Building upon `https://gitlab.com/ada23/guess.git` we now tackle a bit more challenging problem (`https://en.wikipedia.org/wiki/Jotto`) this projectlet. Cursory review suggests the same techniques could work for this case. Further analysis brings up several subtle differences however.

With numbers, each position ie digit has 10 choices. With letters on the other hand we have a lot more 26 if we are talking about **english**. And only some possibilities are **words** with other permutations invalid. Thankfully `https://github.com/dwyl/english-words` has made a wordlist available that we can use.

A further simplification we will make is to restrict the word lengths. Instead of arbitrary word lengths, we will choose a number eg 4 as the length of words involved. It is not surprising that there are over 15000 words of length 5 while the default world length of 4 still gives a choice of well over 7000 words.

The most significant difference in this projectlet is the scoring. With the **cows and bulls** project, the score was a pair - indicating the appearance of a digit and the appearance in the correct position. In the case of **jotto** there is only one score - indicating how many of the characters are found in the word to be guessed. In other words each letter (even if duplicate) contributes 1 to the score if it appears in the word to guess - regardless of the position.

Another major difference between the two is the fact that the scoring for jotto is not **commutative** (Ref: `https://mathworld.wolfram.com/Commutative.html`). It is then imperative that the implementation take into account the order of comparison very carefully.

1.1 Learning Objectives

Dynamic memory allocation and deallocation

2 Specifications

This game is implemented in both modes where the computer is the **dealer** as well as the **player** mode where the human thinks of the number and the computer attempts to guess the number. In theory the computer should be able to arrive at the correct soluion in 5 guesses or less. This project uses a naive approach to the solution which is reasonable for this project size ie 4 digits. It may be prudent to search for better alternatives if the size of the problem is larger.

2.1 Design

Dealer mode The dealer mode is of course trivial. The computer simply uses a time dependent random number to **think** of a number and it is the user who has to guess.

Player mode In the implementation at hand, a simplistic approach is taken in the **player** mode. As the user scores each **guess**, the program just eliminates the numbers from the list of potential solutions. Thus each score reduces the potential solution size - eventually leading up to just one potential (and the real) solution.

2.2 Memory Leak analysis

Enabling the instrumentation and inspecting the results - there are no surprises:

3 Implementation

Implementation in **Ada** `https://gitlab.com/ada23/jotto.git`